Original scientific paper
https://doi.org/10.33180/InfMIDEM2023.205

Informacije MIDEM
Journal of Microelectronics,
Electronic Components and Materials
Vol. 53, No. 2(2023), 103 – 117

Towards Smaller Single-point Failure-resilient Analog Circuits by Use of a Genetic Algorithm

Žiga Rojec

Department EDA, Faculty of Electrical Engineering, University of Ljubljana, Slovenia

Abstract: Failure-resilient analog circuits are difficult to design, but artificial intelligence can help crawl the topology solution space. Using evolutionary computation-based topology synthesis we evolve analog arcus tangent computational circuits, resilient to any rectifying diode or resistor high-impedance single failure or removal. We encode analog circuit topologies as individuals with an upper-triangular incident matrix. Circuits are evolved using a combined technique utilizing parts of NSGA-II and PSADE, based on a special three-dimensional robustness function. We show that topology size for a failure-resilient circuit can be classes smaller than hand-made component-redundancy-based solutions. Our best failure-resilient topology comprises six diodes, three resistors, and a voltage offset source.

Keywords: analog circuits, analog circuit synthesis, circuit optimization, failure-resilience, circuit robustness

Manjšanje analognih vezij odpornih na odpoved poljubne komponente z uporabo genetskega algoritma

Izvleček: Analogna vezja, ki so odporna na napake, je težko načrtovati. Pri prečesavanju prostora možnih topologij lahko pomaga umetna inteligenca. Z sintezo topologij, temelječi na evolucijskem algoritmu, smo razvili analogno računsko vezje za inverzni tangens, ki je odporno na visokoimpedančno okvaro posamezne komponente (diode ali upora) ali njene odstranitve. Topologija analognega vezja je v algoritmu zapisana v obliki zgornje-trikotne vpadne matrike. Vezja razvijemo z uporabo kombinirane metode z uporabo večkriterijskega optimizacijskega algoritma NSGA-II in PSADE, kjer je za usmerjanje sinteze razvita posebna tri-kriterijska funkcija robustnosti. V članku prikazujemo kako zmanjšati velikost topologije, odporne na odpoved komponente, na razrede manjšo velikost od ročno izdelanih robustnih topologij, ki temeljijo na redundanci posameznih komponent. Naš najboljši rezultat je analogno računsko vezje za inverzni tangens, ki je sestavljeno iz šestih diod, treh uporov in odmičnega napetostnega vira.

Ključne besede: analogna vezja, sinteza analognih vezij, optimizacija vezij, odpornost na napake, robustnost vezij

** Corresponding Author's e-mail: ziga.rojec@fe.uni-lj.si*

1 Introduction

Design of an analog circuit is a challenging task, especially when the product has to meet high standards and fulfill tough requirements.

Designers often use various simulation tools to predict temperature, humidity, and electromagnetic behavior during circuit operation. Furthermore, to predict the blueprint manufacturability and maximize the production yield, they also use statistical methods, such as Monte Carlo analysis [1].

However, customer requirements might get even harder. When a device is targeted for use in harsh conditions (i. e., space exploration, aeronautical missions, automotive, robotics), we expect the product to be robust against extreme temperature swings, high ionizing and electromagnetic radiation levels, high working currents, and more. That kind of stress can lead to component faults and premature device failure. Furthermore, failed components in remote and unmanned missions could not be replaced easily.

How to cite:
Ž. Rojec, "Towards Smaller Single-point Failure-resilient Analog Circuits by Use of a Genetic Algorithm", Inf. Midem-J. Microelectron. Electron. Compon. Mater., Vol. 53, No. 2(2023), pp. 103–117

Researchers have already focused on hardening electronic devices against failures per se [2]. The classical ways of doing that include component redundancy, overdesign, shielding and insulation, thermal management, and so on. Most of the time such solutions significantly increase the size, weight, and finally, the cost of the device. The upper methods usually aim to protect every circuit component as if it was the main breaking point of the system.

Researchers have already proposed systems resilient to failures that occur in vivo. Meaning, the circuit has the ability to persist functional when one or more components fails during the circuit operation [3]–[6]. Such systems usually utilize duplicated circuit modules to form redundant sub-systems which are controlled by various voting mechanisms [3], [7]. However, the demultiplexer then becomes the weak part of the system.

This paper shows an alternative method of evolving failure-resilient analog circuits. Using an intensive evolutionary search, we can find novel analog circuit topologies that exhibit robustness to *any* electronic component (semiconductor diode or resistor) high-impedance failure or removal, without a dedicated active demultiplexing system.

We show in this work, that by using an evolutionary topology synthesis tool, we can greatly reduce the size and the number of needed components to achieve failure-resilience of an analog circuit, compared to canonical hand-made design.

To the best of our knowledge, this is one of the few published works on the automated synthesis of a priori robust, failure-resilient nonlinear computational analog circuits [3], [4], [8]–[15], and also one of the first attempts of redundancy reduction by using evolutionary search.

The paper is organized as follows. We summarize previous work on robust topology synthesis in Section 1.1 and describe our motivation in 1.2. We describe the applied topology synthesis technique in Section 2. Results are given in Section 3, summarized in 3.8 and concluded in Section 4.

1.1 Previous work

The discovery of novel circuit topologies has been done by hand for over a century. This is changing with the availability of novel tools, relying on artificial intelligence [16]. Since the beginning of this research area [17]–[19], computer-aided circuit synthesis has become human-competitive and trustworthy for fabrication [16], [20]. We believe, rather than replacing a human expert in the industry, AI might help in the rapid exploring of undiscovered topology space, thereby helping and speeding up the design process.

Reviews of existing analog circuit synthesis techniques can be found in existing literature [21], [22]. However, we give a brief overview of existing topology synthesis efforts for extremely robust and failure-resilient analog circuits below.

1.1.1 Synthesis method

Analog topology synthesis is an extremely non-linear and complex task, which is why most existing approaches in this field search topology with a method, based on the Darwinian selection of the fittest, i.e. evolutionary or genetic algorithm.

Somehow special are the works of Zebulum and Keymeulen, et. Al., who presented an evolutionary algorithm that is being run on the controlling unit of the circuit under failure, in vivo [4], [12].

Evolutionary methods demonstrate a capacity to tackle unconventional challenges. One compelling reason that supports the continued relevance of evolutionary computation, even when compared to neural networks like GNNs, is that they do not always require prior training to align with the defined cost function.

However, emerging tools rooted in GNNs, like CktGNN, showcase impressive capabilities in generating robust circuit topologies [23].

1.1.2 Synthesis goals and degrees of robustness

Passive filters are usually the entry point for showing the performance of analog circuit synthesis tools. Most of the works on failure-resilience also experimented with the synthesis of robust passive analog filter circuits, dealing with various degrees of component faults. Resistor/capacitor/inductor removal was considered in [9], [15], while in addition [3], [7] also studied the complexity of partial and full short-circuit and high-impedance faults. Studies [24]–[27] only considered R/L/C parameter perturbation without full component failure.

Other authors reported syntheses of

- compensator circuit [8] and
- inverter, amplifier, and oscillator [13] resilient to bipolar transistor removal,
- PID controller with R/L/C removal resilience [10],
- transistor-fault resilient amplifier [11],
- half-wave rectifier, NOR gate, and voltage-controlled oscillator for extreme temperature swings (in situ evolution) [12]

- XNOR gate, analog multiplier, and inverter resilient to arbitrary faults in the controlling unit FPTA (Field Programmable Transistor Array) [4]
- the natural logarithm and square-root analog computational circuits resilient to semiconductor diode short-circuit or high-impedance malfunction [28]

1.2 Motivation

1.2.1 Failure-resilience

For this work, let us define failure-resilience as an analog circuit topology property, where any of the basic components (diode or resistor) can be removed or replaced with high-impedance failure, with the circuit showing minimal-to-zero deformation of nominal signal processing abilities. The voltage source and the 10 k Ω input-pullup resistor are excluded from the definition.

The methodology incorporates various failure scenarios using specialized "failure-defining" Spice models, as demonstrated in our prior work [28], where we successfully synthesized analog circuits resilient to both high-impedance and short-impedance failures in semiconductor diodes. In this paper, we primarily concentrate on minimizing topologies that are fully resilient to high-impedance failures. However, due to high computational costs, we do not address short-circuit failures for all component types in this paper; this topic is left for future research.

1.2.2 Size of failure-resilient circuits

Failure-robustness comes with a cost. It is generally paid by (often significantly) higher total number of needed components for the same nominal task as a non-robust circuit would perform. For a system to survive such rigorous change, as one or any component removal/failure, redundant elements and connections must be available in the system.

Let us consider an example of a non-linear, computational analog circuit from Figure 1. The circuit outputs an inverse tangent of input voltage signal between 0 and 10 V. It is a hand-designed linear voltage divider, with diodes used to switch between five linear segments, which closely interpolate the mathematical function [29]. Due to its simplicity, the topology is often used instead of the amplifier-chain summing circuit. If any of the components in the dotted square (except for the voltage source) fails (or is removed), the circuit's transfer function severely changes as seen in Figure 2 with absolute error range plot and Figure 3 with relative error plots.

The most common and straightforward approach to achieving failure-resilience property is to introduce redundancy on a single-component level. In the case of an arctan circuit, every diode has to be paired in parallel and every resistor has to be (at least) tripled in parallel. Two diodes in parallel give a sub-circuit where, theoretically, any of the two diodes might enter high-impedance failure without transfer function transformation. Single resistor with resistance R_n has to be replaced with three parallel resistances 3 R_n to maintain 33% relative error of sub-circuit in case of one resistor entering high-impedance failure.

Figure 4 shows a hand-designed topology that fulfills the failure-resilience criteria. Fair nominal response and narrow error range in failure cases are presented in Figure 5 and Figure 6. Evidently, the circuit topology hence the number of needed components goes off-scale. While the nominal non-robust topology includes 10 resistors and 5 diodes (excluding the input resistor, see 1.2.1), the hand-made robust version comprises 30 resistors and 10 diodes. In CMOS technology, for example, resistors occupy large chip areas [30]. In addition, those resistances are multipliers of the nominal values, which further multiplies the needed area for fabrication. The circuit total cost would be above comparison to the nominal non-robust version.

However, novel studies of analog topology synthesis imply, that number of needed components for failure-resilience might somehow be lower than expected in hand-made designs [3], [7]. The possible reason for that phenomenon is that open-ended topology synthesis allows component-level redundancy to be replaced with system-level redundancy.

1.2.3 Topology size as a synthesis constraint

In this study, we explored the lower limits of topology size for a failure-resilient computational analog circuit. We show, that for the arcus-tangent circuit, the topology could be reduced from 40 critical components in hand-made design down to 8 components by evolutionary-based synthesis. This also has fewer components than used hand-made non-robust design (15).

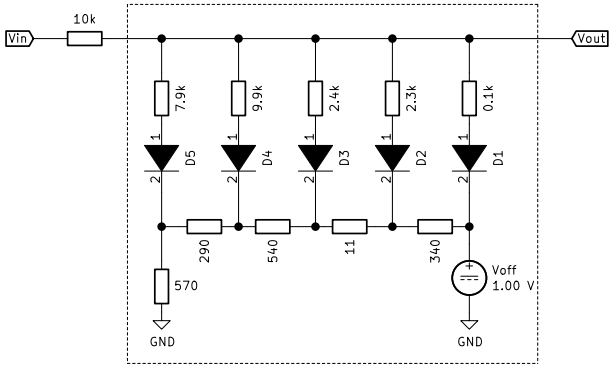

Figure 1: Canonical hand-designed piece-wise linear arctan computational circuit topology.

Our study provides step-by-step size-reducing results for further investigation and a better understanding of underlying mechanisms.

Primary contribution of this paper lies in the demonstration of a novel application of evolutionary methods, resulting in the attainment of system robustness that has not been observed in any existing systems or circuits within the literature.

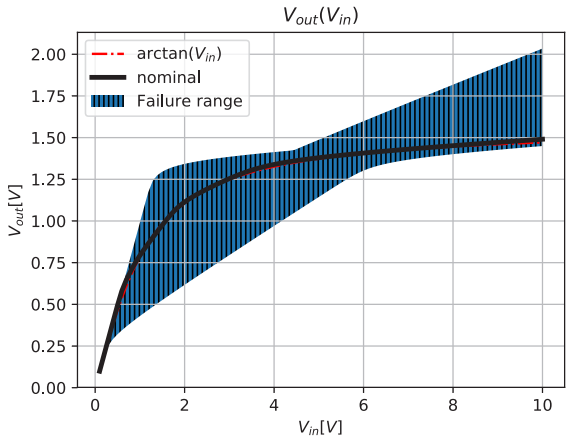

Figure 2: Hand-designed non-robust arctan circuit: nominal response (black) completely covers the arctan function. The range of various failure responses is given in blue.

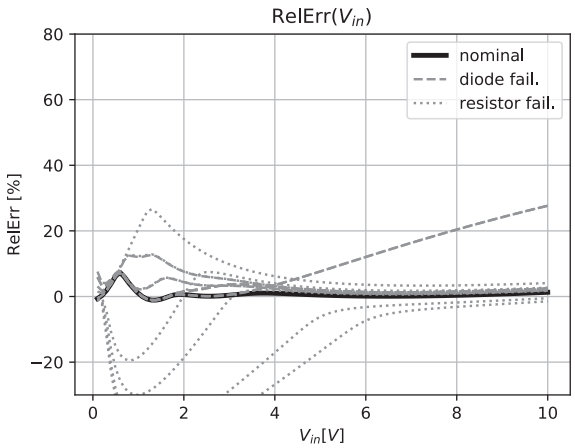

Figure 3: Relative error curves of nominal (solid) and component failures (dotted and dashed).

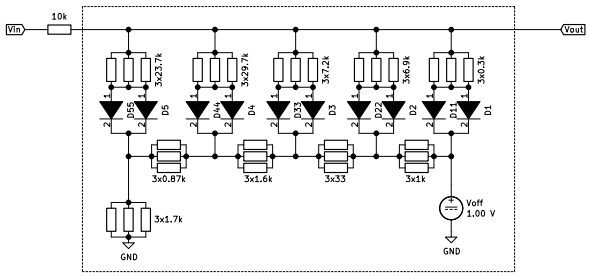

Figure 4: Hand-designed piece-wise linear arctan computational circuit, robust to any single component high-impedance failure or removal.

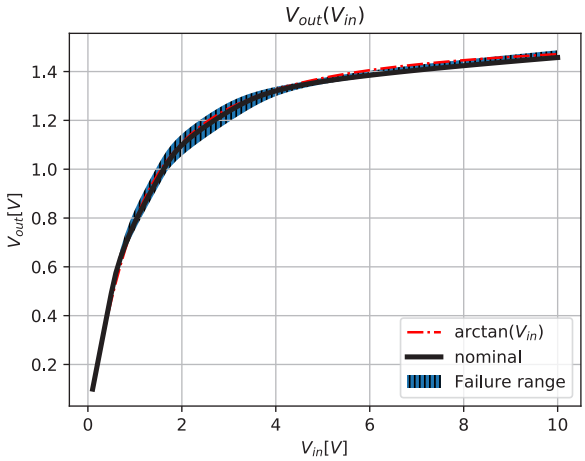

Figure 5: Hand-designed failure-resilient arctan circuit: nominal response (black) covers the arctan function. The range of various failure responses is given in blue.

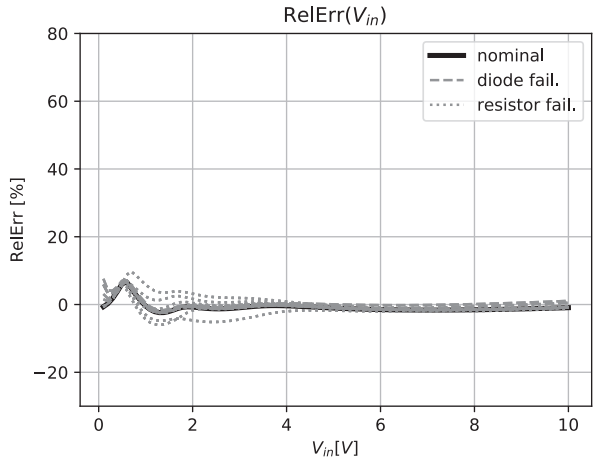

Figure 6: Hand-designed failure-resilient arctan circuit: relative error curves of nominal (solid) and component failures (dotted and dashed).

2 Methods

In this section, we provide details of the methods used in this circuit synthesis. The applied approach is mostly based on [28].

2.1 Analog Circuit Representation

Upper-triangular incident matrix is a well-proven method of encoding an analog circuit topology [22], [28], [31]. It is based on a fixed set of available component terminals. Each building block can comprise one or more input/output terminals (see Figure 7). Usually, the building-block terminals are located on the left side of the fixed set, and outer connections are located on the right-side of the set. The set is then mirrored in two dimensions, forming a connection matrix, where the logical one represents an existing zero-impedance

connection between the terminals on both axes. The matrix is filled with logical ones on a diagonal so that by definition, every terminal is connected to itself. Only the upper matrix triangle is used to exclude half of the redundant mirror connections from the bottom triangle, to reduce the effective matrix size, without sacrificing any topology search space [31], [32]. Additionally, in the inner-connections sector of the matrix, we allow every possible connection, while in the outer-connection section only one positive logical value is allowed per line, filtering-out any connections between outer terminals.

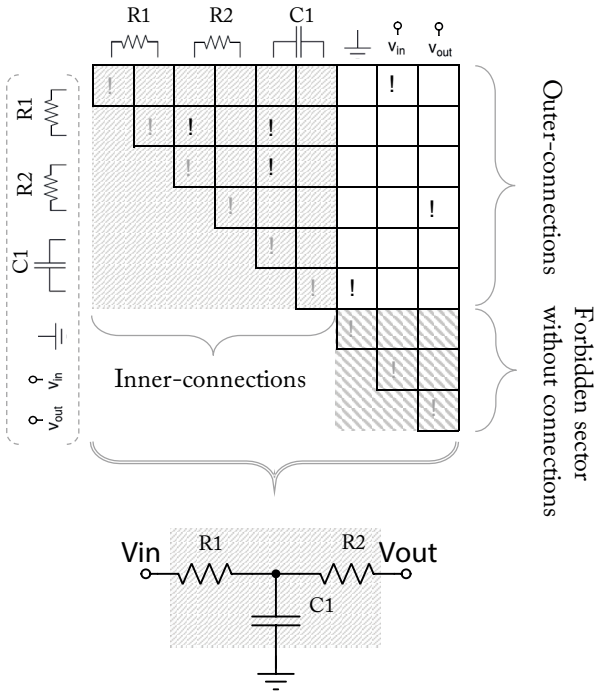

Figure 7: An example of an upper-triangular matrix, representing a simple T-shaped analog circuit topology [31].

Components with adjustable parameters (i.e., resistances, capacitances, transistor widths and lengths, etc.) have their values organized in a separate array, called value vector. While the topology matrix is purely binary, the value vector is a numeric entity.

2.2 Genetic Reproduction and Sizing

For evolutionary computation and mimicking natural genetic reproduction, we use the topology-matrix crossover technique, described in [31]. Every terminal is connected to other terminals via the logical values that reside on a column and a row, intersecting the diagonal element, that represents the connection to itself. By exchanging the two lines of the matrix with another topology matrix, the information of the terminal connecting with the rest of the circuit is transferred. Figure 8 shows two examples of newly-created offspring with one terminal (N=1) and three terminal (N=3) information being exchanged. Note that in the applied algorithm, the number of exchanged terminal connections N is a randomly-chosen number from the set {1,2,3}.

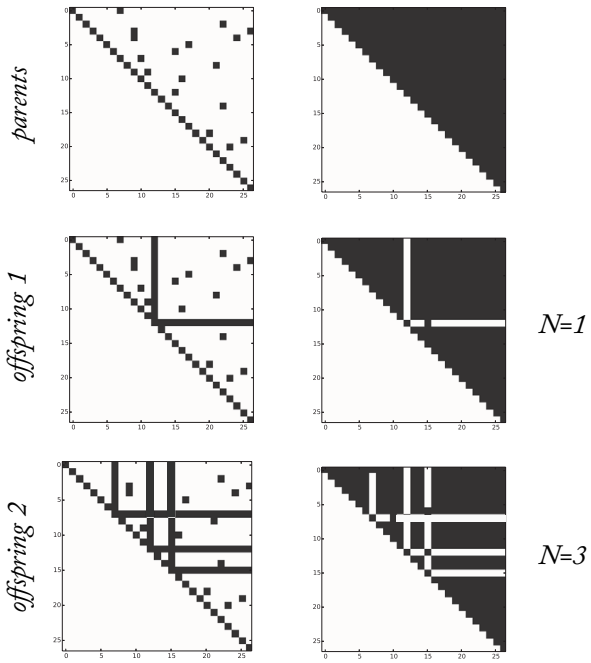

Figure 8: Topology crossover examples. For better illustration, parent no. 2 is a full upper-triangular matrix [31].

The value vector is being optimized using two different methods. The first one is a reproduction mechanism, inspired by a well-known intermediate crossover [33]. The choice between topology-matrix or value-vector crossover is initiated by the evolutionary algorithm. In one case offspring will inherit a modified topology and in another a modified parameter.

Another parameter tuning technique in this work is an established PSADE (Parallel Simulated Annealing and Differential Evolution) [34]. Due to its computational expensiveness (yet effectiveness), it is triggered only every 10th generation on one to three best individuals.

2.3 Fitness function

The fitness function should encompass the desired properties of the circuit. Additionally, it should filter out individuals with unwanted properties and help to guide the searching algorithm through the valley of local minima. We will briefly review the applied fitness function below, but the full justification of chosen criteria is given in [28].

In the case of open-ended topology synthesis, the fitness function definition is rather complex and com-

prises several stages. The first is an evaluation of the circuit's transfer function, i.e. signal processing quality, using a DC analysis in Spice simulator. In the case of arctan circuit design (let us denote the mathematical function as *g*) we calculate the root mean square error (RMSE) between V_{out} (V_{in}) and $g(V_{in})$. We call the result *fitness* and denote it as *f*.

Calculation of failure-resilient circuit fitness needs to be carried out for every predicted failure scenario. In our work, failure-resilience is defined as the high impedance failure of any resistor or semiconductor diode (see 1.2.1). In the case of 30 resistors and 10 diodes, the total number of RMSE calculations must be 41 – that is one for nominal (no failure) scenario f_{nom}, and 40 for every critical device failed, multiplied by the number of failure types considered (only one failure type in this case). Vector ***f*** comprises all RMSE results:

$$\boldsymbol{f} = \left[f_{nom}, f_{1,1}, \ldots, f_{1,F}, \ldots, f_{N,F} \right] \tag{1}$$

where N is the total number of critical components and F is the total number of failure types [28].

Failure-resilient circuit evaluation is carried out in multiple dimensions, and forms a three-dimension robustness vector ***r***:

$$\boldsymbol{r} = \begin{bmatrix} f_{nom} \\ f_{\max} \\ \sigma_f \end{bmatrix} \tag{2}$$

where f_{nom} is RMSE result of no-failure, nominal circuit topology, f_{max} is the maximum of vector ***f*** and α_f is the standard deviation of the same vector [28]. Vector ***r*** gives insight into a single failure-resilient candidate

- nominal performance
- performance in case of worse single-point failure and
- statistical failure scattering.

This separation gives a chance to the NSGA-II algorithm to non-dominantly sort the individuals into Pareto-fronts and by that maintain the genetic diversity, thus avoiding premature convergence.

In the specific case of a failure resilient circuit synthesis, a practitioner might encounter a *false-robustness* phenomenon, which we explain below.

Let us consider an example of a simple diode half-wave rectifier (Figure 9, left). If D0 fails or is removed, the rectifier is no longer working, and statistically, one critical component (diode) makes a 100% chance of circuit failure. Imagine a topology modification, that would harden the circuit against the D0 removal or high-impedance failure. Let us have four additional diodes to fulfill that requirement (one would be enough, but we assume the search algorithm does not know that). The search algorithm can encounter a topology with four diodes with no effect on the nominal transfer function (example in Figure 9 (right)). Still, if D0 fails, the circuit does, too. However, if any of D1-4 fails, the circuit still delivers the transfer function. It appears as only 20% of critical components (diodes) cause a fatal scenario for the circuit. The latter circuit might get promoted because of its better "robustness" value. Obviously, this is not the case, because D1-4 are not electrically connected and do not play any role in signal processing. That kind of circuit has to be ranked out since it does not contribute to real circuit robustness.

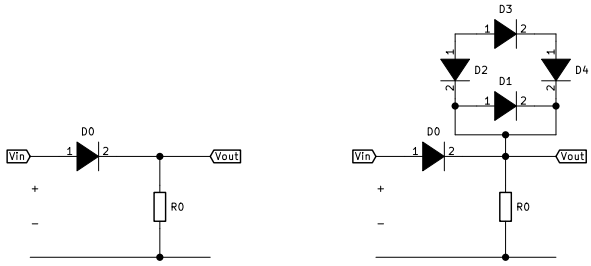

Figure 9: False-robustness problem [28].

Inclusiveness [28] successfully unfolds the false-robustness problem. Using modified diode models and SPICE simulator commands we determine which of the components are electrically connected (*included*) and have an effect on signal processing. Inclusiveness (denoted by *I*) is calculated as a ratio between the number of all critical and included components. Having an updated robustness definition:

$$\boldsymbol{r} = \begin{bmatrix} f_{nom} \\ f_{\max} \\ \sigma_f \end{bmatrix} I \tag{3}$$

circuits with greater inclusiveness are promoted over the circuits with floating or flawed connected components. However, this can lead the synthesis to build larger circuits with excessive redundancy, so component number limits must be set elsewhere in the algorithm. In our case, the top number of available devices is set in the pre-defined component set, which also defines the topology-matrix size. Note that only the inclusiveness of diodes was considered in our work.

2.4 Synthesis algorithm

The search and sorting algorithm utilize major ideas from NSGA-II [35].

The evolutionary algorithm is initiated by a randomly generated population. Then every individual is evaluated according to the fitness/robustness from Section 0. Sorting is performed in three steps, following NSGA-II. In the first step, individuals that do not dominate each other (are not beaten in any combination of objectives) are assigned to a front (i.e. Pareto front). The remaining individuals are put in a second, third, etc., front, with the same non-dominance criteria. A new generation assembly is the second step. We aggregate the new generation starting with individuals from the 1st front, and continue with available individuals from further fronts. Because a union of parents and offspring is usually larger than available space in the new generation, there is a front of individuals, that does not fit as a whole to the new generation. A selection between non-dominated individuals needs to be undertaken. This is done in the third step, the crowding distance calculation. The crowding distance is the distance between two neighboring points (i.e. individuals) along each of the objective axes. Ranking individuals with higher crowding distance helps to a more even distribution in a front of individuals.

After the assembly of the new generation, a parent selection process takes place. With the tournament, some randomly selected individuals are chosen from the generation. The selected individuals compete based on their front number (lower is better) and crowding distance (higher is better). Two tournaments take place to choose two future parents.

Having selected two parents, their genetic material gets reproduced. This can be done by mating their genetic material as in 0 or by mutating it. Control over mating/mutation is a statistical probability, set at the beginning of the algorithm. Similarly, a probability parameter controls whether the topological or parametric part of the gene will be mated/mutated.

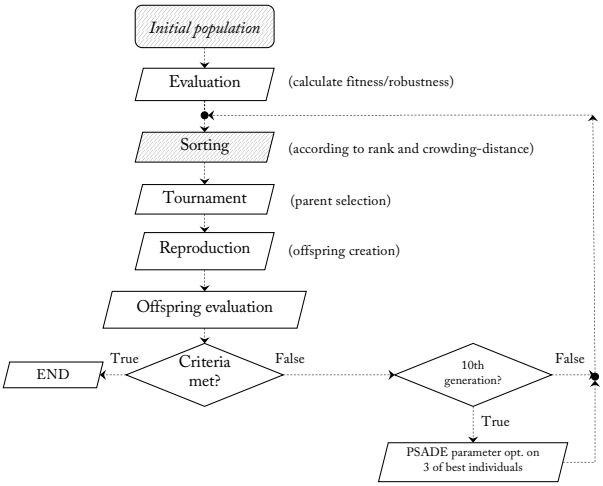

Figure 10: The applied evolutionary algorithm flowchart [31].

We repeat the synthesis algorithm until at least one of the stopping criteria (i.e., design requirements, max. number of generations, timeout.) is met. When ten generations have passed, we run a PSADE [34] parameter optimization on three of the best circuits from the population and thus fine-tune the ambitious individuals.

Figure 10 summarizes the main synthesis algorithm steps.

2.5 Finding minimal topology

Our objective was to evolve circuits with consistent performance even if devices are removed. Initially, we aimed to incorporate as many "redundant" components as possible. However, circuit size doesn't always reflect actual functional contributions, leading to "dummy" or electrically connected but non-functional components.

To address this, we introduced "Inclusiveness" to prevent circuits dominated by dangling sub-circuits, enhancing evolutionary outcomes. Individuals with greater inclusiveness measure propagate more effectively. Our experimentation revealed a paradox when maximizing redundancy while minimizing circuit size simultaneously. Hence, we perform separate stages for minimizing and maximizing circuit schematics. We are listing two more reasons, why the size of circuit schematics is not another objective of NSGA-II search.

Our topology representation method using an upper-triangular incident matrix limits arbitrary extensions during evolution runs. Varying matrix sizes in the evolutionary pool cause inconsistent crossovers and mating patterns.

The third concern relates to the computational complexity of NSGA-II and evaluating circuits under different failure scenarios. A variable maximum component number during evolution would increase computational effort, impacting NSGA-II's performance and circuit robustness evaluation. As a result, we chose not to experiment with variable component numbers to minimize computational burden.

3 Results

Our experiment comprised eight independent topology searches. For each synthesis we predefined the set of available components, that is N_d diodes and N_r resistors that are subject to possible high-impedance failure. V_{off} and a R_{in} input resistor (the latter was non-optional) were also available with each synthesis but were excluded from failure consideration.

The main part of the experiment was discovering the possibilities of finding topologies with fewer components than in hand-designed examples (e.g., from Figure 4), that perform arcus tangent analog calculation and exhibit the failure-resilience property (1.2.1).

The genetic algorithm parameters were fixed through the experiment and are summarized in Table 1.

Table 1: Genetic algorithm properties.

Parameter	Value
Population	1000
Tournament	3
Mating prob.	0.6
Topology reproduction prob.	0.8

Resistance values were limited to the range between 10 and 100 kΩ, and voltage source with DC range of 0 to 6 V. Every synthesis was conducted on an i9 HP desktop, utilizing 16 computational threads on 8 processor cores.

3.1 Synthesis with a max of 12 diodes, 12 resistors

With the ambition to cut the number of needed components for the circuit, we gave the first upper limit of Nd_{max} = 12 and Nr_{max}=12. This is already a significant cut of the total number of components (Nd + Nr) in comparison to hand designed example from Figure 4 which comprises 40 components. The algorithm can, however, synthesize a topology with fewer elements.

Starting with a random population, without any prior knowledge available in the population itself, we let the combined NSGA-II algorithm run for 306 generations

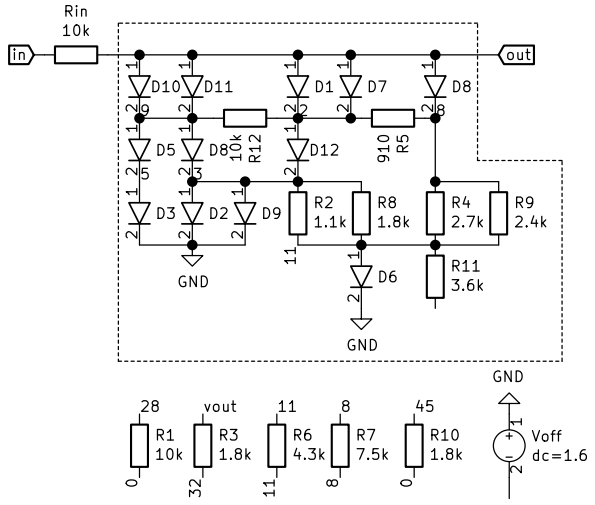

Figure 11: Synthesized arctan computational circuit (Nd_{max} = 12, Nr_{max} = 12), robust to any single component high-impedance failure or removal.

(roughly 15 hours). The outcome is presented in Figure 11. The final topology comprises all 12 available diodes. Some resistors were excluded from the final topology since they do not have any signal-processing effect (such as short-connected resistors, or resistors connected to simulator-helper nodes). The voltage source was also not included in the final design. We excluded some of the components already from topology schematics in Figure 11.

We summarize the circuit performance in three parameters: nominal topology RMSE is 0.312, the worst failure RMSE is 0.370 and the standard distribution of all cases (nominal and failures) is 0.026. One can visualize those results in Figure 12 and Figure 13.

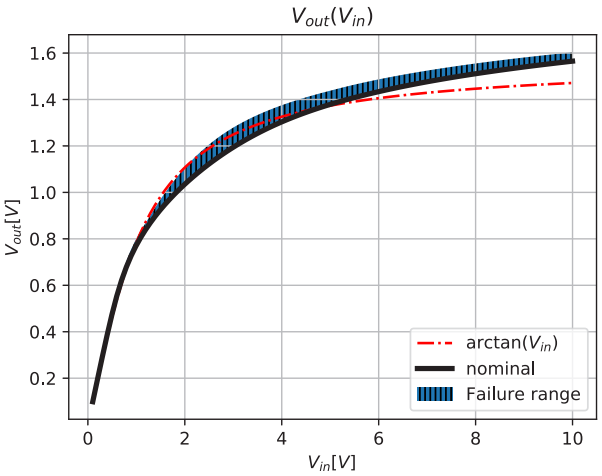

Figure 12: Synthesized arctan computational circuit (Nd_{max} = 12, Nr_{max} = 12): nominal response (black), arctan function (red, dashed-dotted). The range of various failure responses is given in blue.

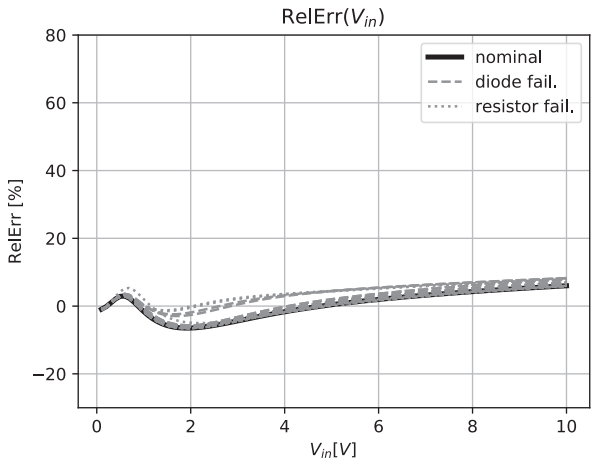

Figure 13: Synthesized arctan computational circuit (Nd_{max} = 12, Nr_{max} = 12): relative error curves of nominal (solid) and component failures (dotted and dashed).

Together with a voltage source, six available resistors were not used in the final circuit. That is why we con-

ducted our experiment with tighter device component limits.

3.2 Synthesis with a max of 10 diodes, 10 resistors

The next synthesis was limited to Nd_{max} = 10 and Nr_{max}=10. We stopped the algorithm after 822 generations (that was after 33h).

The outcome is presented in Figure 14. The final topology comprises all 10 available diodes. Two resistors were not included in the final topology.

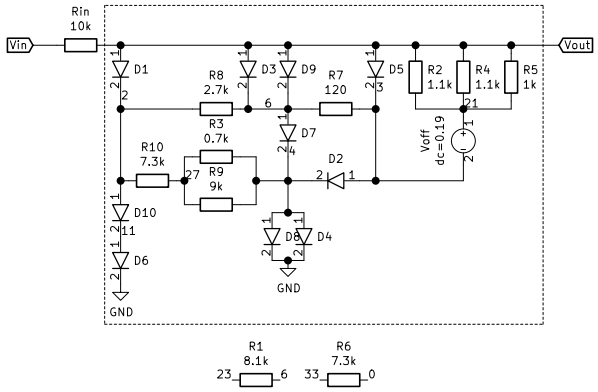

Figure 14: Synthesized arctan computational circuit (Nd_{max} = 10, Nr_{max} = 10), robust to any single component high-impedance failure or removal.

Circuit performance: nominal topology RMSE is 0.158, the worst failure RMSE is 0.270 and the standard distribution of all cases (nominal and failures) is 0.032. One can visualize failure ranges in Figure 15 and Figure 16. This circuit performs better than the one from the previous synthesis, according to the three observables. It also comprises 2 diodes less and four resistors more.

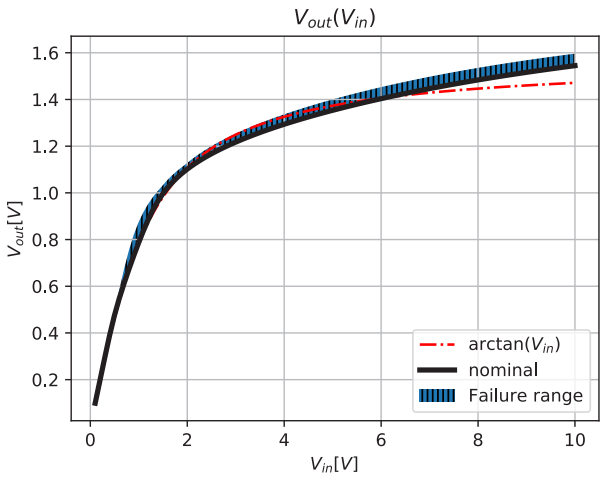

Figure 15: Synthesized arctan computational circuit (Nd_{max} = 10, Nr_{max} = 10): nominal response (black), arctan function (red, dashed-dotted). The range of various failure responses is given in blue.

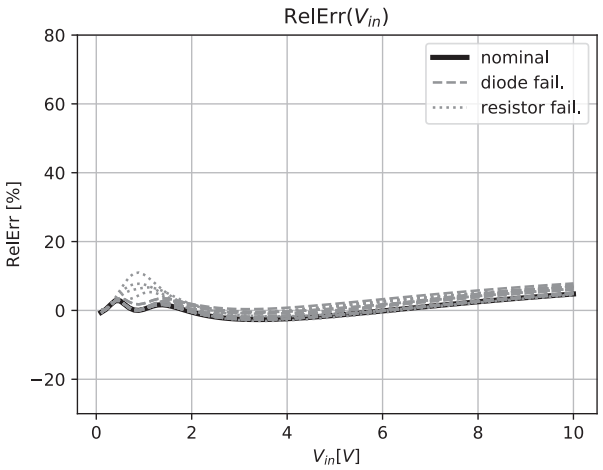

Figure 16: Synthesized arctan computational circuit (Nd_{max} = 10, Nr_{max} = 10): relative error curves of nominal (solid) and component failures (dotted and dashed).

3.3 Synthesis with a max of 8 diodes, 8 resistors

We proceed with Nd_{max} = 8 and Nr_{max} = 8. We stopped the algorithm after 432 generations (11h).

The outcome is presented in Figure 17. The final topology comprises 6 diodes and 6 resistors that can fail during the circuit operation. Two resistors and two diodes were not included in the final topology.

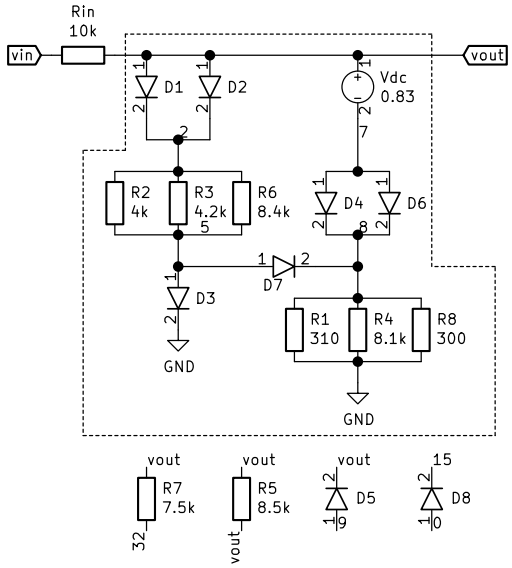

Figure 17: Synthesized arctan computational circuit (Nd_{max} = 8, Nr_{max} = 8), robust to any single component high-impedance failure or removal.

Circuit performance: nominal topology RMSE is 0.149, the worst failure RMSE is 0.152 and the standard distribution of all cases (nominal and failures) is 0.017. One can visualize failure ranges in Figure 18 and Figure 19.

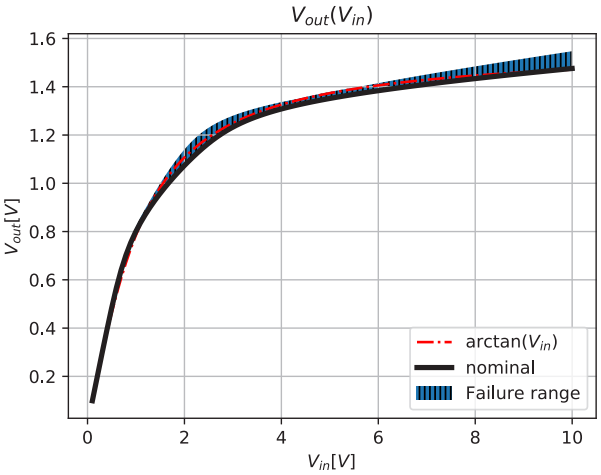

Figure 18: Synthesized arctan computational circuit (Nd_{max} = 8, Nr_{max} = 8): nominal response (black), arctan function (red, dashed-dotted). The range of various failure responses is given in blue.

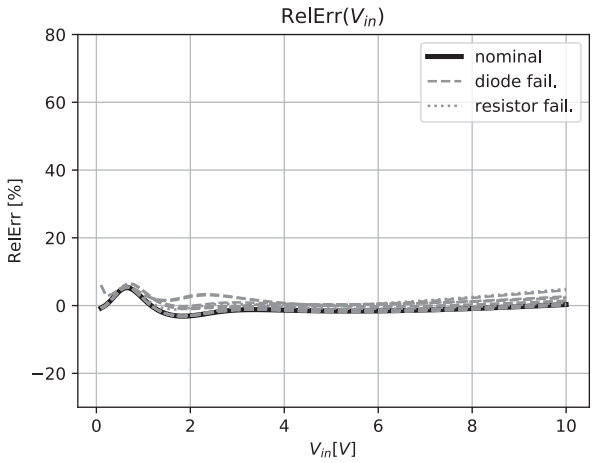

Figure 19: Synthesized arctan computational circuit (Nd_{max} = 8, Nr_{max} = 8): relative error curves of nominal (solid) and component failures (dotted and dashed).

Because the algorithm kept solving the problem using less than the maximum of available components, we proceed and further tighten the Nd_{max} and Nr_{max} criteria.

3.4 Synthesis with a max of 6 diodes, 6 resistors

We stopped the Nd_{max} = 6 and Nr_{max}= 6 synthesis after 2340 generations (48 h).

Figure 20 shows the outcome. The final topology uses all available diodes and three out of six available resistors.

Circuit performance: nominal topology RMSE is 0.106, the worst failure RMSE is 0.110 and the standard distribution of all cases (nominal and failures) is 0.008. One can visualize failure ranges in Figure 21 and Figure 22.

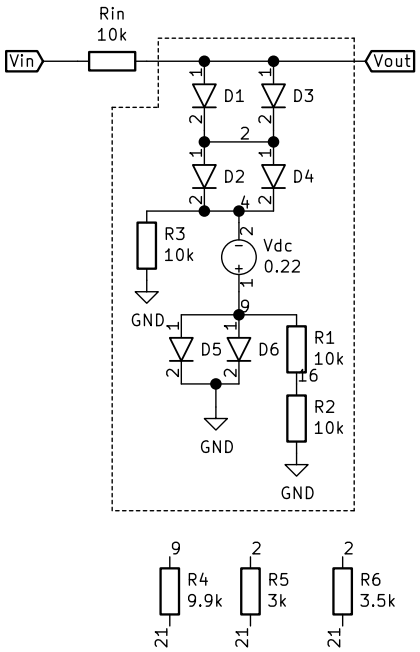

Figure 20: Synthesized arctan computational circuit (Nd_{max} = 6, Nr_{max} = 6), robust to any single component high-impedance failure or removal.

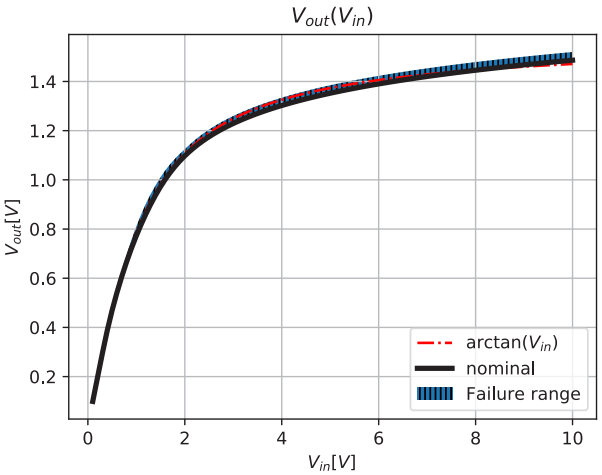

Figure 21: Synthesized arctan computational circuit (Nd_{max} = 6, Nr_{max} = 6): nominal response (black), arctan function (red, dashed-dotted). The range of various failure responses is given in blue.

3.5 Synthesis with a max of 5 diodes, 5 resistors

The Nd_{max} = 5 and Nr_{max}= 5 synthesis was stopped after 2582 generations (36 h).

As shown in Figure 23, the final topology comprises all available components.

Although the synthesis comprises only ten critical components (plus voltage source and input resistor), the performance was not yet diminished. The nominal

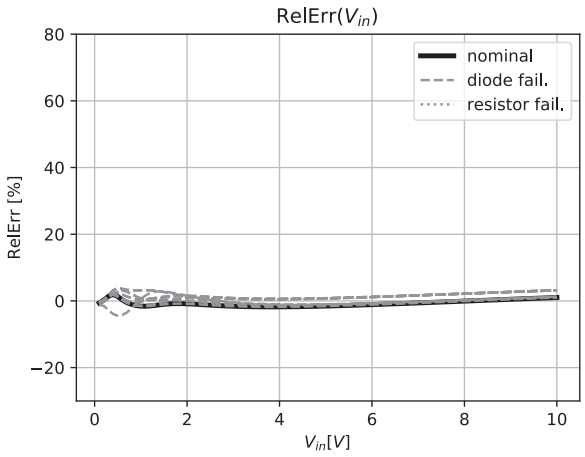

Figure 22: Synthesized arctan computational circuit (Nd_{max} = 6, Nr_{max} = 6): relative error curves of nominal (solid) and component failures (dotted and dashed).

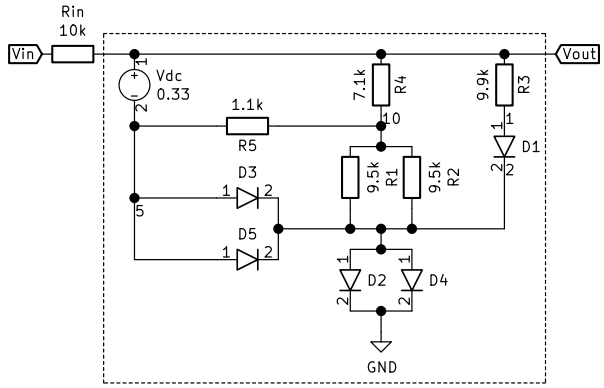

Figure 23: Synthesized arctan computational circuit (Nd_{max} = 5, Nr_{max} = 5), robust to any single component high-impedance failure or removal.

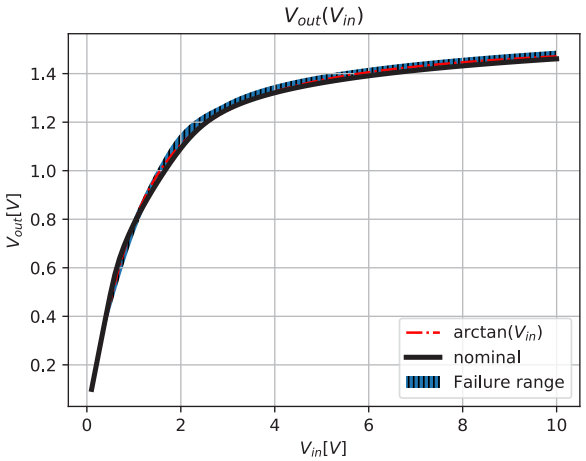

Figure 24: Synthesized arctan computational circuit (Nd_{max} = 5, Nr_{max} = 5): nominal response (black), arctan function (red, dashed-dotted). The range of various failure responses is given in blue.

topology RMSE is 0.108, the worst failure RMSE is 0.165 and the standard distribution of all cases is 0.022. See failure ranges in Figure 24 and Figure 25.

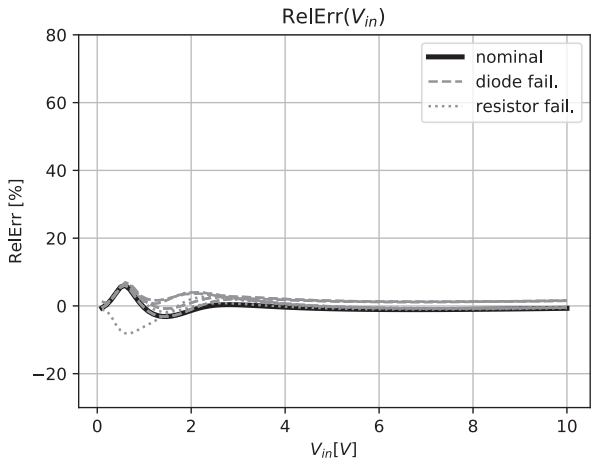

Figure 25: Synthesized arctan computational circuit (Nd_{max} = 5, Nr_{max} = 5): relative error curves of nominal (solid) and component failures (dotted and dashed).

3.6 Synthesis with a max of 4 diodes, 4 resistors

Searching for the bottom limit, we conducted the Nd_{max} = 4 and Nr_{max} = 4 synthesis. We finished it after 1077 generations and 12h.

The final topology comprised 4 resistors and 4 diodes (Figure 26).

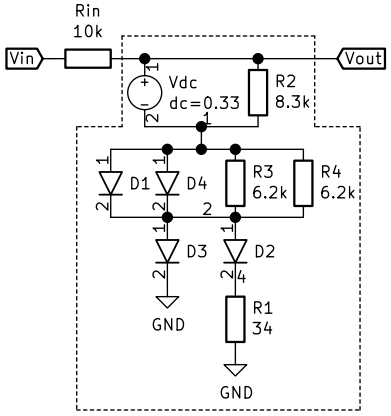

Figure 26: Synthesized arctan computational circuit (Nd_{max} = 4, Nr_{max} = 4), robust to any single component high-impedance failure or removal.

The nominal topology RMSE is 0.173, the worst failure RMSE is 0.217 and the standard distribution of all cases is 0.028. See failure ranges in Figure 27 and Figure 28.

We have discovered, that this synthesis is a probable bottom limit in our experiment. To illustrate, how a smaller design poorly fits the requirement, we show one more synthesis.

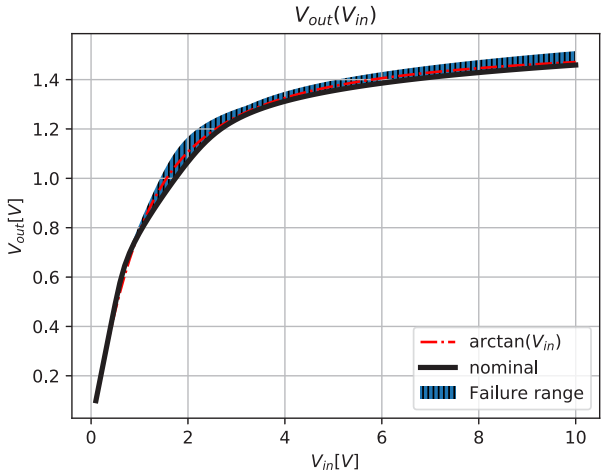

Figure 27: Synthesized arctan computational circuit (Nd_{max} = 4, Nr_{max} = 4): nominal response (black), arctan function (red, dashed-dotted). The range of various failure responses is given in blue.

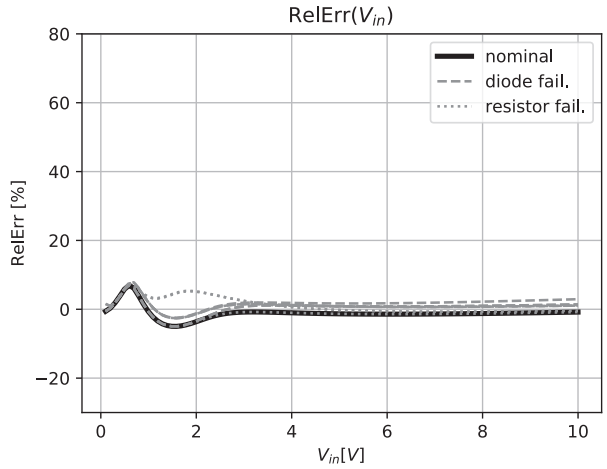

Figure 28: Synthesized arctan computational circuit (Nd_{max} = 4, Nr_{max} = 4): relative error curves of nominal (solid) and component failures (dotted and dashed).

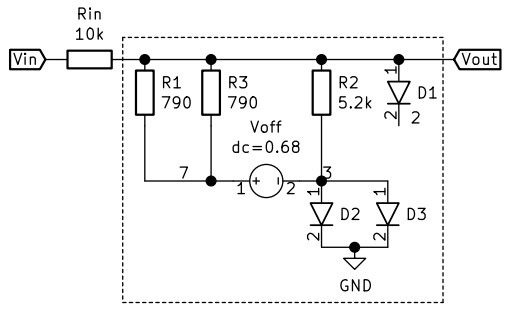

Figure 29: Synthesized arctan computational circuit (Nd_{max} = 3, Nr_{max} = 3), robust to any single component high-impedance failure or removal.

3.7 Synthesis with a max of 3 diodes, 3 resistors

Using limits Nd_{max} = 3 and Nr_{max}= 3 synthesis, we finished the search after 3188 generations (11h).

See Figure 29 for the topology. The nominal topology RMSE is 0.497, the worst failure RMSE is 0.507 and the standard distribution is 0.010. Failure ranges are shown in Figure 30 and Figure 31. We can observe a two-piece approximation of the arctan function, which yields high RMSE.

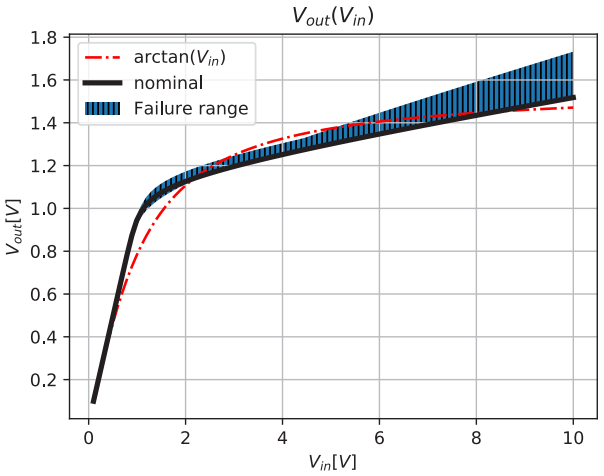

Figure 30: Synthesized arctan computational circuit (Nd_{max} = 3, Nr_{max} = 3): nominal response (black), arctan function (red, dashed-dotted). The range of various failure responses is given in blue.

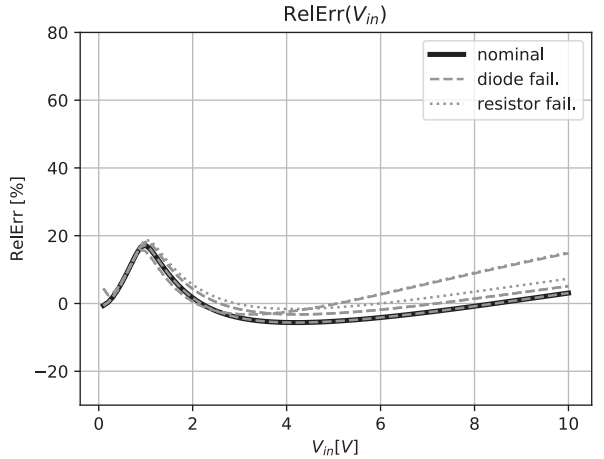

Figure 31: Synthesized arctan computational circuit (Nd_{max} = 3, Nr_{max} = 3): relative error curves of nominal (solid) and component failures (dotted and dashed).

3.8 Result Summary

Table 1 summarizes the experiment results. Surprisingly, tightening the number of available diodes and resistors has led to improved circuit performance in

both nominal functionality and robustness, with its best at Nd=6, Nr=3. Although initial syntheses involved searches over $Nd_{max} > 6$, $Nr_{max} > 3$ topology space, the Nd = 6, Nr = 3 best solution was not discovered in these.

Table 2: Results of a conducted experiment. Every row is an independent topology synthesis with different num. of component limits. The first row is the hand-made robust design.

Nd_{max}	Nr_{max}	Nd	Nr	f_{nom}	f_{max}	σ_f
N/A	N/A	10	30	0.116	0.262	0.047
12	12	12	6	0.312	0.370	0.026
10	10	10	8	0.158	0.270	0.032
8	8	6	6	0.149	0.152	0.017
6	6	6	3	**0.106**	**0.110**	**0.008**
5	5	5	5	0.108	0.165	0.022
4	4	4	4	0.173	0.217	0.028
3	3	2	3	0.497	0.507	0.010

There might be several reasons for that phenomenon. The first, most obvious one, is an enormous search space for topology search. Within one synthesis run, we cannot sample every possible circuit, but rather crawl the space using the evolutionary search. This is why two evolutionary syntheses with the same goal but different initial settings might not produce the same outcome.

The second reason is more related specifically to the robustness definition in our experiment. As noted, our problem definition does not reward circuits with fewer components, but rather the opposite. Inclusiveness (see 2.3) rewards circuits that electrically include all available components to push means of redundancy into the circuit and avoid false robustness. During the synthesis, while the objectives might already be met with requirements, the inclusiveness criteria might draw the search toward more included components, which makes the search too wide and lasting long. We conclude, that with such-defined search problem, the hard limits on the topology size and the number of available components are key to an efficient small-size failure-resilient topology search.

4 Conclusions

Using the topology synthesis tools, we can find topologies, that exhibit novel properties, such as failure tolerance. We showed that failure-resilience in analog circuits can be achieved with smaller-than-expected topologies, by introducing system-level redundancy instead of much more expensive component-level redundancy. Using an evolutionary-based topology synthesis tool, we introduced novel topologies of analog arcus tangent circuit. The most compact one comprises six diodes, three resistors, a voltage source, and an input resistor. Each of the diodes and the three resistors can fail or be removed, with almost no computational error.

Based on this research, we can conclude that the integration of system redundancy for single-point failures was achieved by imposing a strict limitation on the maximum size of available components. We showed, that to achieve such resilience, surprisingly low number of electrical components is needed.

In the realm of CMOS design, reducing the number of components doesn't necessarily translate to cost savings on its own. However, we conducted a brief analysis of the total resistance for both robust circuits, encompassing both hand-crafted and synthesized designs. Total resistance can provide a rough estimate of circuit area in certain CMOS processes. For instance, the total resistance of a hand-designed circuit (as shown in Fig. 4) amounts to approximately 219 kΩ, whereas the resistance of the best synthesized circuit totals around 20 kΩ (a difference of a decade).

Furthermore, reducing the number of components can have a direct impact on cost savings in the realm of discrete electronics, such as PCBs. In the domain of discrete resistors, the resistance value itself does not significantly affect the cost of the device, assuming factors like manufacturer, package, power rating, and tolerance remain the same. With this in mind, the minimization of robust topologies emerges as a pivotal factor in achieving cost-effective and highly reliable circuits.

In comparison to previous experiments, this study considers not only diodes, but also resistors to be a possible point of failure. We experimented with evolutionary search for circuits that are robust to both, short-circuit and open-circuit failures in all possible failure points (components), including some experiments including transistors. However, we acknowledge that further investigation and modified approaches are required to address this specific problem effectively.

We believe our work will inspire further practitioners in the field of analog circuit topology synthesis.

5 Supplementary material

The source code of the synthesis tool is available online at https://github.com/zigarojec/MatrixCircEvolutions.

6 Acknowledgments

I would like to thank my colleagues from the EDA department of the Faculty of Electrical Engineering, the University of Ljubljana for all the support in my work.

The authors acknowledge the financial support from the Slovenian Research and Innovation Agency (research core funding No. P2-0246 ICT4QoL—Information and Communications Technologies for Quality of Life).

7 Conflict of Interest

We can declare no conflict of interest in this work.

8 References

1. Á. Bűrmen in H. Habal, „Computing Worst-Case Performance and Yield of Analog Integrated Circuits by Means of Mesh Adaptive Direct Search“, *Inf. MIDEM*, let. 45, str. 160–170, jun. 2015.
2. Y. Deval, H. Lapuyade, in F. Rivet, „Design of CMOS integrated circuits for radiation hardening and its application to space electronics“, v *2019 Ieee 13th International Conference on Asic (asicon)*, F. Ye in T. A. Tang, Ur., New York: Ieee, 2019. Pridobljeno: 10. avgust 2022. [Na spletu]. Dostopno na: https://www.webofscience.com/wos/woscc/full-record/WOS:000541465700105
3. M. Liu in J. He, „An Evolutionary Negative-Correlation Framework for Robust Analog-Circuit Design Under Uncertain Faults“, *Ieee Trans. Evol. Comput.*, let. 17, št. 5, str. 640–665, okt. 2013, https://doi.org/10.1109/TEVC.2012.2228208.
4. D. Keymeulen, R. S. Zebulum, Y. Jin, in A. Stoica, „Fault-tolerant evolvable hardware using field-programmable transistor arrays“, *Ieee Trans. Reliab.*, let. 49, št. 3, str. 305–316, sep. 2000, https://doi.org/10.1109/24.914547.
5. M. Xue in J. He, „Evolutionary topology programming for analog circuit fault tolerant design“, v *2013 25th Chinese Control and Decision Conference (CCDC)*, maj 2013, str. 3391–3396. https://doi.org/10.1109/CCDC.2013.6561534.
6. S. Askari, M. Nourani, in A. Namazi, „Fault-tolerant A/D converter using analogue voting“, *IET Circuits Devices Amp Syst.*, let. 5, št. 6, str. 462–470, nov. 2011, https://doi.org/10.1049/iet-cds.2011.0042.
7. K.-J. Kim, A. Wong, in H. Lipson, „Automated synthesis of resilient and tamper-evident analog circuits without a single point of failure“, *Genet. Program. Evolvable Mach.*, let. 11, št. 1, str. 35–59, mar. 2010, https://doi.org/10.1007/s10710-009-9085-2.
8. R. S. Zebulum, M. Vellasco, M. A. Pacheco, in H. T. Sinohara, „Evolvable hardware: On the automatic synthesis of analog control systems“, v *2000 IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484)*, mar. 2000, str. 451–463 let.5. https://doi.org/10.1109/AERO.2000.878521.
9. K.-J. Kim in S.-B. Cho, „Combining Multiple Evolved Analog Circuits for Robust Evolvable Hardware“, v *Intelligent Data Engineering and Automated Learning - IDEAL 2009*, E. Corchado in H. Yin, Ur., v Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2009, str. 359–367. https://doi.org/10.1007/978-3-642-04394-9_44.
10. G. A. Hollinger in D. A. Gwaltney, „Evolutionary design of fault-tolerant analog control for a piezoelectric pipe-crawling robot“, v *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, v GECCO '06. New York, NY, USA: Association for Computing Machinery, jul. 2006, str. 761–768. https://doi.org/10.1145/1143997.1144133.
11. Q. Ji, Y. Wang, M. Xie, in J. Cui, „Research on Fault-Tolerance of Analog Circuits Based on Evolvable Hardware“, v *Evolvable Systems: From Biology to Hardware*, L. Kang, Y. Liu, in S. Zeng, Ur., v Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2007, str. 100–108. https://doi.org/10.1007/978-3-540-74626-3_10.
12. R. S. Zebulum, A. Stoica, D. Keymeulen, L. Sekanina, R. Ramesham, in X. Guo, „Evolvable hardware system at extreme low temperatures“, v *Evolvable Systems: From Biology to Hardware*, J. M. Moreno, J. Madrenas, in J. Cosp, Ur., Berlin: Springer-Verlag Berlin, 2005, str. 37–45. Pridobljeno: 17. avgust 2021. [Na spletu]. Dostopno na: https://www.webofscience.com/wos/woscc/summary/3e9863eb-c395-495e-94b0-3f857c12151a-048a98b8/date-descending/1
13. P. Layzell in A. Thompson, „Understanding Inherent Qualities of Evolved Circuits: Evolutionary History as a Predictor of Fault Tolerance“, v *Evolvable Systems: From Biology to Hardware*, J. Miller, A. Thompson, P. Thomson, in T. C. Fogarty, Ur., v Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2000, str. 133–144. https://doi.org/10.1007/3-540-46406-9_14.
14. S. Ando in H. Iba, „Analog Circuit Design with Variable Length Chromosomes“, str. 8.
15. K.-J. Kim in S.-B. Cho, „Automated synthesis of multiple analog circuits using evolutionary computation for redundancy-based fault-tolerance“, *Appl. Soft Comput.*, let. 12, št. 4, str. 1309–1321, apr. 2012, https://doi.org/10.1016/j.asoc.2011.12.002.
16. A. Mirhoseini *idr.*, „A graph placement methodology for fast chip design“, *Nature*, let. 594, št. 7862,

str. 207-+, jun. 2021, https://doi.org/10.1038/s41586-021-03544-w.
17. W. Kruiskamp in D. Leenaerts, „Darwin: Analogue circuit synthesis based on genetic algorithms", *Int. J. Circuit Theory Appl.*, let. 23, št. 4, str. 285–296, 1995, https://doi.org/10.1002/cta.4490230404.
18. J. R. Koza, F. H. Bennett III, D. Andre, M. A. Keane, in F. Dunlap, „Automated Synthesis of Analog Electrical Circuits by Means of Genetic Programming", *Trans Evol Comp*, let. 1, št. 2, str. 109–128, jul. 1997, https://doi.org/10.1109/4235.687879.
19. H. Y. Koh, C. H. Sequin, in P. R. Gray, „OPASYN: a compiler for CMOS operational amplifiers", *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, let. 9, št. 2, str. 113–125, feb. 1990, https://doi.org/10.1109/43.46777.
20. T. McConaghy, P. Palmers, M. Steyaert, in G. G. E. Gielen, „Trustworthy Genetic Programming-Based Synthesis of Analog Circuit Topologies Using Hierarchical Domain-Specific Building Blocks", *IEEE Trans Evol. Comput.*, let. 15, str. 557–570, 2011.
21. S. E. Sorkhabi in L. Zhang, „Automated topology synthesis of analog and RF integrated circuits: A survey", *Integration*, let. 56, str. 128–138, 2017.
22. Ž. Rojec, Á. Bűrmen, in I. Fajfar, „Analog circuit topology synthesis by means of evolutionary computation", *Eng. Appl. Artif. Intell.*, let. 80, str. 48–65, apr. 2019, https://doi.org/10.1016/j.engappai.2019.01.012.
23. Z. Dong, W. Cao, M. Zhang, D. Tao, Y. Chen, in X. Zhang, „CktGNN: Circuit Graph Neural Network for Electronic Design Automation", predstavljeno na The Eleventh International Conference on Learning Representations, sep. 2022. Pridobljeno: 8. avgust 2023. [Na spletu]. Dostopno na: https://openreview.net/forum?id=NE2911Kq1sp
24. J. He, K. Zou, in M. Liu, „Section-representation scheme for evolutionary analog filter synthesis and fault tolerance design", v *Third International Workshop on Advanced Computational Intelligence*, avg. 2010, str. 265–270. https://doi.org/10.1109/IWACI.2010.5585181.
25. S. Li, W. Zou, in J. Hu, „A Novel Evolutionary Algorithm for Designing Robust Analog Filters", *Algorithms*, let. 11, št. 3, str. 26, mar. 2018, https://doi.org/10.3390/a11030026.
26. J. Hu, X. Zhong, in E. D. Goodman, „Open-ended robust design of analog filters using genetic programming", v *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, v GECCO '05. New York, NY, USA: Association for Computing Machinery, jun. 2005, str. 1619–1626. https://doi.org/10.1145/1068009.1068283.
27. S. Ando in H. Iba, „Analog circuit design with a variable length chromosome", v *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, jul. 2000, str. 994–1001 let.2. https://doi.org/10.1109/CEC.2000.870754.
28. Ž. Rojec, I. Fajfar, in Á. Burmen, „Evolutionary Synthesis of Failure-Resilient Analog Circuits", *Mathematics*, let. 10, št. 1, Art. št. 1, jan. 2022, https://doi.org/10.3390/math10010156.
29. A. K. Kenneth, „Piecewise Linear Circuits", mar. 2004.
30. F. Maloberti, „Design of CMOS Analog Integrated Circuits".
31. Ž. Rojec, J. Olenšek, in I. Fajfar, „Analog Circuit Topology Representation for Automated Synthesis and Optimization", *Inf. Midem-J. Microelectron. Electron. Compon. Mater.*, let. 48, št. 1, str. 29–40, mar. 2018.
32. G. Györök, „Crossbar network for automatic analog circuit synthesis", v *2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, jan. 2014, str. 263–267. https://doi.org/10.1109/SAMI.2014.6822419.
33. D. G. Tomasz, *Genetic Algorithms Reference*. Tomasz Gwiazda, 2006.
34. J. Olenšek, T. Tuma, J. Puhan, in Á. Bűrmen, „A new asynchronous parallel global optimization method based on simulated annealing and differential evolution", *Appl. Soft Comput.*, let. 11, št. 1, str. 1481–1489, 2011, https://doi.org/10.1016/j.asoc.2010.04.019.
35. K. Deb, A. Pratap, S. Agarwal, in T. Meyarivan, „A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", *Trans Evol Comp*, let. 6, št. 2, str. 182–197, apr. 2002, https://doi.org/10.1109/4235.996017.

Arrived: 17. 02. 2023
Accepted: 06. 10. 2023